

Patent Application of
Ovidiu Feodorov, Iulian Musat, Dimitri Bevc, and Alexander M. Popovici
for

Systems and Methods for Collaboratively Viewing and Editing Seismic Data

RELATED APPLICATION DATA

[0001] This application claims the benefit of the filing date of U.S. Provisional Patent Application No. 60/450,290, filed 02/26/2003, entitled "Systems and Methods for Collaboratively Viewing and Editing Seismic Data," which is herein incorporated by reference.

FIELD OF THE INVENTION

[0002] This invention relates to geophysical prospecting using seismic signals, and in particular to systems and methods for managing and viewing seismic data.

BACKGROUND

[0003] Computer-intensive processing of reflection seismic data is the main tool for imaging the Earth's subsurface to identify hydrocarbon reservoirs and determine rock and fluid properties. Seismic data are recorded at the earth's surface or in wells, and an accurate model of the underlying geologic structure is constructed by processing the data. In the past decade, 3-D seismic processing has shown far superior cost-effectiveness than conventional 2-D seismic processing. However, the reconstruction of accurate 3-D images of the subsurface requires the handling of a huge amount of seismic data (on the order of terabytes for a single modern marine 3-D survey) and the application of computer-intensive imaging algorithms. Even today, only large-scale parallel computers can image modern marine surveys with the most sophisticated algorithms within a useful turn-around time. Indeed, the use of large-scale parallel computers for seismic processing has achieved such technical and economical success that the geophysical market accounts for one of the largest commercial market for scientific high-performance computing.

[0004] Many exploration organizations have no direct access to high-end seismic imaging technologies because they lack the resources to acquire and maintain the necessary hardware

and software. The cost of the required computing and data-storage facilities is still quite high, although it keeps going down. However, in perspective, even more problematic is the large investment in application development and system maintenance that the effective use of these computational facilities requires. These costs are not decreasing, they are actually increasing with the level of complexity and sophistication of the imaging applications. Many existing commercial packages are targeted to run effectively on workstations, not on large parallel systems, and thus are ineffective for the high-end applications. As a result, only the central processing departments of large oil companies can afford to employ large-scale parallel computers for imaging 3-D seismic data. The rest of the exploration industry outsources large 3-D seismic imaging projects to outside service companies.

[0005] According to this outsourcing model, the data are physically sent on tape to the service companies, who process the data with little input from the oil companies, and then deliver the final images after several months. This service model has been historically successful for exploring areas with relatively simple geology, but it has proven inadequate for exploring areas with complex geology, where most of the still undiscovered hydrocarbon reservoirs are yet to be found. In such areas, the data must often be imaged by applying 3-D prestack depth migration, instead of simpler and more traditional time imaging procedures. Depth imaging is potentially much more accurate than time imaging, but it is also less robust with respect to some of the processing parameters. In particular, depth imaging needs an accurate interval velocity function to properly focus the reflections. Because the velocity function cannot be uniquely determined from the data alone, a priori geological information must be taken into account, and the final results are greatly enhanced when geologists work closely with the processing team. However, in the typical interaction between service companies and oil companies, the geologist cannot be involved in the processing because of the physical distance and the difficulties caused by the old traditional service model.

SUMMARY

[0006] In the preferred embodiment, a computer-implemented seismic viewing/editing collaboration method includes performing collaborative cursor tracking, copaging, picking, and image manipulation in a distributed-display-processing, peer-to-peer architecture. A

parameterized, minimal set of information required to update a display is transferred directly between different clients.

DESCRIPTION OF THE FIGURES

[0007] Fig. 1 is a schematic diagram of a preferred hardware environment of a seismic processing system of the present invention.

[0008] Fig. 2 is a flowchart illustrating a collaborative seismic processing method according to a preferred embodiment of the present invention.

[0009] Fig. 3 is a flowchart illustrating a collaborative method including collaborative cursor tracking, copaging, picking, and image manipulation, according to a preferred embodiment of the present invention.

[0010] Fig. 4 shows an exemplary screen shot of a 2D viewer, including a display of two users' cursors, according to a preferred embodiment of the present invention.

[0011] Fig. 5 is a flowchart illustrating a collaborative picking method according to a preferred embodiment of the present invention.

DETAILED DESCRIPTION

[0012] In the following description, a set of elements is understood to comprise one or more elements. It is understood that results and other data structures can be generated and/or stored in system memory and/or in files. Geophysical data encompasses recorded as well as synthetic data. Unless otherwise characterized, geophysical data processing is understood to broadly encompass any transformation, generation, visualization, use or accessing of geophysical data. An action performed automatically in response to an event is an action performed without requiring user intervention after the event. Unless otherwise limited, the term multicasting encompasses sending data to one or more recipients from a selected group, and is distinct from broadcasting, which involves sending data to any user having available equipment.

[0013] The following description illustrates the present invention by way of example and not necessarily by way of limitation.

[0014] Fig. 1-A shows a schematic overview of a preferred hardware environment of a geophysical data processing system 20 of the present invention. System 20 comprises a server subsystem 22 and a plurality of client subsystems 24. Each client subsystem 24 is connected to server subsystem 22 and to the other client subsystems 24 over corresponding control and interpretation (analysis) links 26, which are preferably implemented over a wide area network such as the Internet. One or more of links 26 can also be implemented over a local area network (LAN). Client subsystems 24 can be located far away (for example within a different building or company) than server subsystem 22. Links 26 include client-server links connecting clients 24 to server 22, as well as client-client links interconnecting clients 24 directly (independently of server 22).

[0015] Server subsystem 22 preferably comprises a processing unit 30 and a data storage device 32. Processing unit 30 can include an integrated parallel computer such as an Origin 2000 from SGI or Enterprise from Sun Microsystems. Processing unit 30 can also include a cluster of interconnected computers, such as for example a Linux cluster forming a parallel computer. Data storage device 32 includes an array of tape and/or hard disk drives capable of storing geophysical data sets--on the order of hundreds of gigabytes to tens of terabytes of data. Processing unit 30 is connected to data storage device 32 over a data link 34. Processing unit 30 and storage device 32 are preferably located within the same physical facility, and can be implemented on the same computer system. Data link 34 can have a higher-bandwidth than control/interpretation links 26, and can be implemented over a local area network (LAN). Computationally-intensive tasks such as processing performed on seismic data sets (e.g. migration, velocity analysis, datuming) are preferably performed on server 22.

[0016] Each client subsystem 24 can be implemented using a personal computer or workstation with less storage and/or computational capability than server subsystem 22. Each client subsystem 24 includes processing, storage, and display and input devices (e.g. monitor

and keyboard) for control, data entry, and data visualization. The client storage capabilities are preferably used to store synchronized local copies of seismic data sets used to generate 3D and 2D image displays, and associated data such as pick sets. The locally stored images can range in size from hundreds of kB to tens or hundreds of MB. Each client subsystem **24** includes a graphical user interface (GUI) and an associated geophysical data viewer. The data viewer preferably includes a 3D Viewer for displaying and manipulating 3D images, and a 2D Viewer for displaying and manipulating 2D images such as slices of a 3D image. Different client subsystems **24** may have different operating systems (platforms). The GUI and geophysical data viewer are preferably implemented in a platform-independent language such as Java. In one implementation, the GUI is run as an applet in a conventional browser, although in general the GUI can be provided as a standalone installed program. Both server **22** and clients **24** may have the Java Runtime Environment (JRE) installed thereon.

[0017] Clients **24** may be used to remotely control geophysical data processing performed on server **22**, for example as described in U.S. Patent No. 6,493,635, herein incorporated by reference. In addition, collaboration engines running on clients **24** also allow a geographically-distributed team of geophysicists to work together on a project/dataset. A processing project can include performing several data processing iterations over a common set of data files, pick files, and data processing flows. The collaboration engines preferably allow different members of the team to visualize, manipulate and edit in real time geophysical data, picks, and other parameters. During a collaboration session, the collaboration engines running on clients **24** can communicate directly with each other, without the intermediation of server **22**, as described below. Clients **24** are also capable of retrieving data from server **22** at the beginning of a collaboration session, and of saving data on server **22** during a collaboration session or when the session is over.

[0018] At least some applications running on clients **24** may be connected to server **22** and to each other through a network protocol implemented using the Remote Method Invocation (RMI) application programming interface (API). Other applications may employ TCP or UDP connections. The Java Remote Method Invocation system allows an object running on one Java Virtual Machine (VM) to invoke methods on an object running on another Java VM.

The system can be used to implement an authentication protocol based on the Java Authentication and Authorization Service (JAAS). A Secure Socket Layer (SSL) protocol can also be implemented selectively below the JAAS level, and on top of TCP/IP, for communications outside a secure zone such as a corporate Intranet. Code loaded on clients **24** and server **22** may be signed, i.e. assigned permissions based on a security policy currently in effect. The security policy can specify permissions available for given code, users, and locations.

[0019] Fig. 2 is a flowchart illustrating a seismic collaboration method **100** according to an embodiment of the present invention. A method such as the one shown in Fig. 2 may be implemented on each client **24**. In a first step **102**, a user employs the GUI of a client **24** to log on to server **22**. Client **24** then retrieves project information from server **22**, including a list of available collaboration rooms, as shown at **104**. A project includes a set of seismic data files, a set of flow (processing) files, and a set of collaboration rooms. A project may include additional associated files. Each project can contain recursively an arbitrary number of sub-projects. A project group or workgroup can include multiple projects with associated user permissions, for example multiple projects of the same customer.

Collaboration Room

[0020] In a step **106**, the user creates a new collaboration room or joins an existing collaboration room. A collaboration room is a virtual space maintained primarily on server **102**, where a number of users meet and generate input that ultimately translates into quasi-concurrent changes applied to the data which is the subject of the collaboration. Each collaboration room can be graphically represented as an object in a project tree structure implemented as described in the above-incorporated U.S. Patent No. 6,493,635. Preferably, a collaboration room has a specific icon, and is a subelement of the site tree root, of a project group or of a project. The hierarchical position of a collaboration room in the site defines the user permissions required to join the collaboration room. For example, all users may access a site collaboration room not associated with a project or project group. Only users with permission to access a specific project group or project can use the corresponding collaboration rooms. Different projects or project groups may correspond to different

customers or groups within a customer company. A given project or project group can include multiple collaboration rooms, and a user may be a member of multiple collaboration rooms.

[0021] In a step 108, client 24 retrieves from server 22 a list of users and event generators the users/clients contributed to the collaboration room. The users present at a given moment in a collaboration room can be graphically represented as subelements of the room node in the tree structure. Once a user connects to a room and becomes member of that collaboration room, the user's name is made public and becomes available to the other members of the room. Preferably, a member has permanent access to the list of the other members participating in the collaboration, and is able to make two choices relative to any other member: whether to send events to the other member, and whether to receive/accept events from the other member. By default, all events (described below) are sent and received to/from all members.

Collaboration Event Generators

[0022] Preferably, available event generators include a chat interface, a 2D data viewer, and a 3D data viewer. The event generators are preferably run on each client 24, and the event generators communicate directly with other clients 24. The event generators may also communicate with server 22, for example to retrieve data at the beginning of a session and to save data at the end of a session. The chat interface is a part of the GUI allowing a collaboration room member to send and receive text events to/from other members. The 2D data viewer is a module specialized in displaying 2D geophysical images. The 2D data viewer preferably provides a number of tools including zoom, color map control, copaging, and picking. Events sent and received by a 2D data viewer preferably include cursor position tracking, copaging, image modification, and picking events. The 3D data viewer is a module specialized in displaying 3D geophysical images. The 3D data viewer preferably provides a number of tools including zoom, color map control, viewing angle control, and picking. Events sent and received by a 3D data viewer include cursor position tracking, projection transformation data, and picking events. Projection transformation data is information needed to apply a transform to the 3D data to generate a 2D image to be displayed on a monitor, for a given viewing angle and distance.

[0023] Preferably, events are divided into two categories: group state (stateful) events and transitory events. Group state events include image manipulation, copaging, and pick set manipulation events. Groups state events modify the group state on each client **24**, and the group state is enforced to be synchronized between different clients **24**. Preferably, the loss of state events in the transmission process is prevented completely, since such a loss would lead to desynchronization of the group state and would affect subsequent state data. Preventing data loss can be achieved through the use of a reliable multicast protocol which verifies the receipt of data, such as a protocol implemented using the open source JGroups (Javagroups) API. Delivery of a stateful event is complete only when all recipient clients have confirmed receipt of the event. Transitory events include cursor position tracking events. Loss of some transitory event data is acceptable. For transitory events, maintaining a high-throughput and low latency is preferably more important than preventing data loss. Transitory events are preferably transmitted using a transmission protocol does not verify receipt of data.

[0024] The transmitted events are parameterized object descriptions (object parameters), rather than full graphical descriptions (pixel sequences). A set of object parameters includes, for example: a cursor/client identity, two cursor coordinates, and a cursor color. A client receiving the object parameters can construct a graphical representation of the object (e.g. cursor) locally. A full graphical description of a cursor would include an array of pixels defining an image area containing the graphical representation of the cursor.

[0025] A user using a client **24** may select an existing event generator from the list retrieved from server **22**, as shown at **110** in Fig. **2**. Choosing an existing event generator causes the event generator of the selected type to start running on client **24**, and communicate with event generators of that type on other clients **24**. A user may also start a new event generator (step **112**) and enable collaboration for that event generator (step **114**). Enabling collaboration leads to the transfer of an event generator identifier to server **22** for addition to the event generator list accessible by other clients **24**. Starting or selecting an existing event generator such as a data viewer can involve transferring a data set including seismic data and associated other data (e.g. pick set, velocity model) from server **22** or one client **24** to another client **24**.

[0026] In a step 118, multiple clients 24 perform collaborative viewing, manipulation, and editing of geophysical data and associated data. The collaborative process includes sending and receiving events directly to/from other clients 24, as described in further detail below. Each client 24 updates its display to reflect the actions performed by other clients 24 in real time, e.g. with an average latency of less than 1 s, preferably on the order of tens to hundreds of ms, ideally on the order of ms (i.e. less than 10 ms). In practice, it was observed that latencies on the order of a few ms are commonly achieved for clients collaborating from locations in Houston, TX, and Silicon Valley, CA. Once a user decides to end his or her participation in the collaborative process, the user can save the collaboration results on server 22 (step 120), close the event generator(s) running on the corresponding client 24, and leave the collaboration room.

[0027] Fig. 3 is a flowchart illustrating an exemplary collaborative data viewing method 200 according to a preferred embodiment of the present invention. A user employing a client 24 joins a collaboration room (step 206), and retrieves from server 22 a data set for the collaboration room including a list of users and data viewers contributed by the users to the collaboration room (step 208). The user can then choose an existing data viewer (step 210), or start a new data viewer (step 212) and enable collaboration for that data viewer (step 214). An exemplary screenshot of a data viewer according to an embodiment of the present invention is shown in Fig. 4.

[0028] The user can then engage in collaborative viewing, manipulation, and editing of geophysical data and associated data, as shown generally at 218. Preferably, the collaborative process can include collaborative cursor tracking (step 230), copaging (step 232), picking (step 234), and image manipulation (step 236). Steps 230-236 can be performed repeatedly, in parallel and in any order. Once the collaboration process is over, the user can close the data viewer (step 222). Each of the collaborative steps 230-236 is described in further detail below.

Collaborative Cursor Display

[0029] In step 230, a client 24 sends current cursor position data for that client directly to all other clients, and receives current cursor position data from all other clients. The current cursor position includes an identification of the client and two cursor coordinates. Each client 24 then locally generates corresponding displays of the local cursor and the remote cursors, and generates displays of corresponding cursor labels. Fig. 4 shows two exemplary cursor displays labeled "User 2" and "User 3." Each time a cursor is moved by its corresponding user, the corresponding client 24 sends directly to all other clients the current cursor position. Each client 24 updates its cursor display whenever the local cursor or a remote cursor has changed position. The cursor updating is performed in real time. The cursor position data is preferably transmitted directly between different clients 24, rather than indirectly through server 22, in order to minimize the latency of the cursor display process. Generally, the cursor position data may also be transmitted through server 22. The latency of the collaborative display process is also minimized by transmitting only cursor parameters needed to reconstruct each cursor locally, rather than a graphical representation of the cursor. Generating the cursor displays is then performed locally on each client 24 using the parameterized cursor display data.

Collaborative Copaging

[0030] In step 232, a client 24 can change the current 2D page or image displayed by the viewer. The current 2D page is typically a slice within a larger 3D volume comprising a plurality of 2D slices. Client 24 alters the data display to reflect the updated page, and sends updated page identification information directly to all other clients 24. All other clients then automatically update their data display to reflect the current 2D page chosen by the first client 24. Preferably, a user can choose to control co-paging manually, or automatically in a timed sequence.

Collaborative Picking

[0031] In a step 234 shown in Fig. 2, a user can employ a corresponding client 24 to collaboratively change a pick set corresponding to the current image displayed by the 2D viewer. Fig. 5 is a flowchart illustrating a collaborative picking method 300 according to a

preferred embodiment of the present invention. In a first step 340, a client 24 determines whether any collaborative pick set corresponding to a selected data viewer is active. If no pick set is active, a user can direct client 24 to create a new active collaborative pick set (step 342). A pick set is preferably a shared state (e.g. a file or part of a file) stored in a
5 synchronized manner on all clients 24, and associated with a 3D data set. Each pick can correspond to a location within a 2D or 3D image. The pick set preferably includes an ordered list of 3D coordinates for a plurality of picks, an identification of the 3D image to which the picks correspond, and a pick set name. The pick set is displayed by generating graphical representations of pick icons at each location, and of segments sequentially
10 interconnecting the picks in the list. The collaborative pick set is maintained in a common, synchronized shared state on all participating clients 24. Multiple pick sets can be grouped together in a horizon.

[0032] Once a collaborative set is active, any client 24 can add (step 344), move (step 346) or delete (step 348) picks to/from the pick set. Preferably, a user may also change the display
15 color of a pick set. The pick set modification commands are multicast to all other clients 24 in real time. The corresponding display generated by each viewer is updated to reflect the modifications to the pick set, by adding, moving, or deleting pick representations (icons) and associated segments from the corresponding image display. Any user may employ a corresponding client 24 to save the collaborative pick set.

20 Collaborative Image Manipulation

[0033] In a step 236 shown in Fig. 2, a user can employ a corresponding client 24 to collaboratively perform image manipulation operations on the current image displayed by the 2D viewer. The manipulated image is ordinarily a velocity model image. Image manipulation operations can include changing a velocity model displayed according to user commands, for
25 example by moving velocity interface boundaries, filling an area defined between pick sets with a constant velocity, or painting a velocity within a region with a brush tool having a customizable width. A shared state is maintained for the current velocity model on all clients 24 that participate in a collaboration session.

[0034] In the preferred approach described above, parameterized interaction-related tasks are transferred directly between different clients 24, although generally server 22 may also be used as a communication relay, particularly when direct communication between clients 24 is difficult. Server 22 is preferably used only for computationally-intensive geophysical data processing steps such as imaging and velocity analysis, while communication, display generation, picking, co-paging and other tasks are performed locally on each client 24 and are transferred directly to other clients 24. The data transferred is parameterized, minimal meta-information (e.g. cursor coordinates) required to generate the displays by processing performed on each client 24, rather than processed graphical information (e.g. a set of cursor pixel locations). This parameterized, direct communications approach reduces the load on the links between clients 24 and server 22.

[0035] By contrast, in a system employing the X-protocol and its variations (e.g. the X11R6 Broadway extension), the display rendering and other related operations are performed on the server, and each client connects to other clients only through the server. While such a “thin-client” approach allows a conventional server application to be run remotely without editing, the network communications can suffer from high latency. The high latency is a consequence of the relatively large amount of graphical data sent between the server and the clients, and of the centralization of all communications through each client-server link. Collaboration in a system using a “thin-client” approach can be frustrating, particularly as the number of users increases.

[0036] The present invention further provides computers and computer systems programmed to perform the methods described herein, computer-readable media encoding instructions to perform the methods described herein, and computer apparatus comprising means for performing the method steps described herein. Suitable means for performing the method steps described herein include computer hardware programmed with corresponding software.

[0037] It will be clear to one skilled in the art that the above embodiments may be altered in many ways without departing from the scope of the invention. Accordingly, the scope of the invention should be determined by the following claims and their legal equivalents.